

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-152591
 (43)Date of publication of application : 16.06.1995

(51)Int.CI. 606F 9/46
 606F 15/16

(21)Application number : 06-229952 (71)Applicant : SIEMENS AG
 (22)Date of filing : 26.09.1994 (72)Inventor : WITTE MARTIN
 JOERG OEHLERICH
 HELD WALTER

(30)Priority

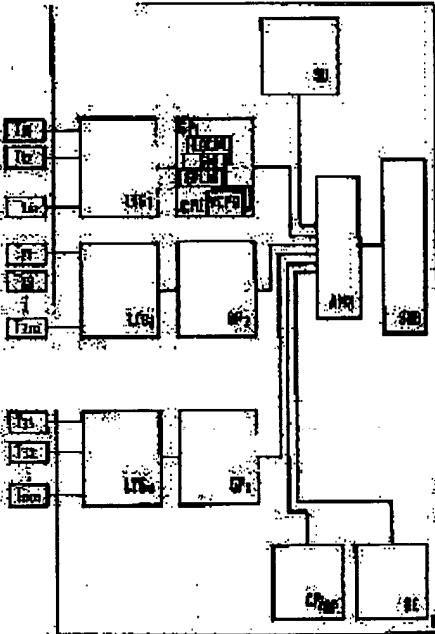
Priority number : 93 93115467 Priority date : 24.09.1993 Priority country : EP

(54) BALANCE ADJUSTMENT METHOD FOR LOAD

(57)Abstract:

PURPOSE: To provide a method for distributing the excessive loads of respective processors to the processor with vacancy without affecting the other load states based on the consideration of the load state of the processor for the balance adjustment method of loads in a multiprocessor system capable of processing generated tasks under real time conditions by a plurality of processors GPn.

CONSTITUTION: The respective processors GPi obtain the load state GPLSi in the form of a quantized amount and the load state GPLSn of the different processor GPn is reported to the respective processors by a time pattern. The respective processors send out at least a part of the tasks generated in the processor to the remaining processors and distribute the sent-out tasks to the remaining processors corresponding to the load states of the remaining processors depending on the fact that the load state exceeds a prescribed mount and depending on the load states of the other processors.



LEGAL STATUS

[Date of request for examination] 06.07.2001

[Date of sending the examiner's decision of rejection] 28.01.2005

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-152591

(43)公開日 平成7年(1995)6月16日

(51) Int.Cl. G 06 F 9/46 15/16	識別記号 3 6 0 B 8120-5B 3 8 0 Z 8219-5L	序内整理番号 F I	技術表示箇所
--------------------------------------	--	---------------	--------

審査請求 未請求 請求項の数10 CL (全 10 頁)

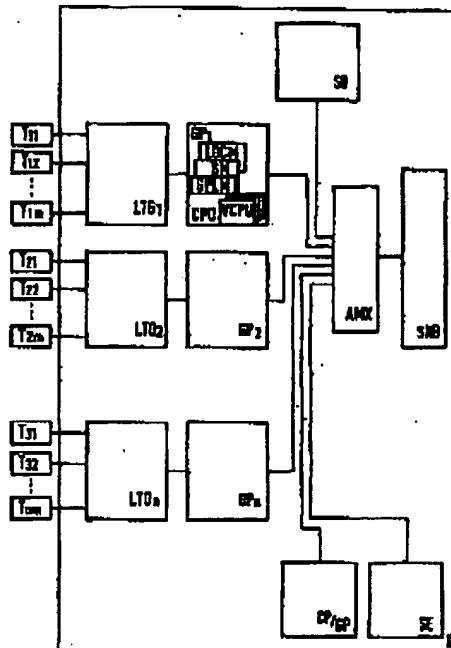
(21)出願番号 特願平6-228952	(71)出願人 シーメンス アクチエンゲゼルシャフト SIEMENS AKTIENGESELLSCHAFT ドイツ連邦共和国 ベルリン 及び ミュンヘン (番地なし)
(22)出願日 平成6年(1994)9月26日	(72)発明者 マルティン ヴィッテ ドイツ連邦共和国 ウルム ヴェッペリン シュトラーセ 7
(31)優先権主張番号 93116467.8	(72)発明者 イェルク エーラーリッヒ ドイツ連邦共和国 ストックドルフ ブレントルヴェーク 1
(32)優先日 1993年9月24日	(74)代理人 弁理士 矢野 敏雄 (外2名)
(33)優先権主張国 ドイツ (DE)	最終頁に続く

(54)【発明の名称】負荷の平衡調整方法

(57)【要約】

【目的】生じるタスクを複数のプロセッサG P nによってリアルタイム条件下で処理することができる、マルチプロセッサシステムにおける負荷の平衡調整方法を、プロセッサの負荷状態の考慮に基づいて、個々のプロセッサの過負荷が、空きのあるプロセッサに、その他の負荷状態に影響することなく分配される方法を提供することである。

【構成】各プロセッサG P i がその負荷状態G P L S i を定量化された量の形において求め、各プロセッサに、別のプロセッサG P n の負荷状態G P L S n を時間パターンで通報し、各プロセッサが、負荷状態が所定の量を上回ることに依存しておよびその他のプロセッサの負荷状態に依存して、プロセッサに生じるタスクの少なくとも一部を前記残りのプロセッサに送出し、該送出されたタスクを、残りのプロセッサの負荷状態に相応して該残りのプロセッサに分配する。



(2)

特開平7-152591

【特許請求の範囲】

【請求項1】 生じるタスクを複数のプロセッサ(GP_n)によってリアルタイム条件下で処理することができる、マルチプロセッサシステムにおける負荷の平衡調整方法において、それぞれのプロセッサ(GP_i)がその負荷状態($GPLS_i$)を定量化された量の形において求め、それぞれのプロセッサ(GP_i)に、別のプロセッサ(GP_n)の負荷状態($GPLS_n$)をある時間バターンで通報し、それぞれのプロセッサ(GP_i)が、当該プロセッサの負荷状態($GPLS_i$)が所定の量を上回ることに依存しておよびその他のプロセッサ(GP_n)の負荷状態に依存して、当該プロセッサに生じるタスクの少なくとも一部を前記残りのプロセッサ(GP_n)に送出し、該送出されたタスクを、前記残りのプロセッサ(GP_n)の負荷状態($GPLS_n$)に相応して該残りのプロセッサに分配することを特徴とする負荷の平衡調整方法。

【請求項2】 プロセッサ(GP_i)は、その負荷状態($GPLS$)が再び前記所定の量を下回る程度の数のタスクしか送出しない請求項1記載の負荷の平衡調整方法。

【請求項3】 負荷状態を特徴付ける前記定量化された量はそれぞれ、その上側並びに下側の領域移行においてヒステリシス値を備えている請求項1記載の負荷の平衡調整方法。

【請求項4】 或る1つのプロセッサ(GP_i)からタスクを残りの複数のプロセッサ(GP_n)に送出する際に、付加的に、前記プロセッサ(GP_i)のタスク(過負荷)防護に対する尺度を決定する請求項1記載の負荷の平衡調整方法。

【請求項5】 プロセッサに関連付けられた負荷状態($GPLS_n$)を、「それぞれがそれぞれに通報する」通信を用いて前記通信システム(K)における前記プロセッサ(GP_n)の個別コントロールマネージャから直接前記プロセッサ(GP_n)のすべての別のコントローラマネージャに分配する請求項1記載の負荷の平衡調整方法。

【請求項6】 プロセッサに関連付けられた負荷状態($GPLS_n$)を所定のプロセッサ(GP_x)のロードコントロールマネージャ(LOCM)において統合する請求項1記載の負荷の平衡調整方法。

【請求項7】 時間バターンに相応して、所定のプロセッサ(GP_x)において統合された負荷状態($GPLS_n$)をその都度、前記プロセッサ(GP_n)の表に記憶する請求項6記載の負荷の平衡調整方法。

【請求項8】 前記プロセッサのその都度の負荷状態($GPLS$)に依存して、当該プロセッサによって付加的にまだ処理可能なタスクの部分($p_i(n)$)を確定する請求項1から7までのいずれか1項記載の負荷の平衡調整方法。

【請求項9】 プロセッサ(GP_i)のその都度の負荷状態($GPLS_n$)に依存して、該プロセッサから分散分配すべきタスク($q(i)$)を高めるかまたは低減する請求項1から8までのいずれか1項記載の負荷の平衡調整方法。

【請求項10】 それぞれのプロセッサにおいて、残りのプロセッサが付加的に引き受けるべきタスクの部分($p(i)$)から成る和結果に依存して、過負荷時における分散分配すべきタスク($q(j)$)の部分を決定する請求項1から9までのいずれか1項記載の負荷の平衡調整方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、生じるタスクを複数のプロセッサ(GP_n)によってリアルタイム条件下で処理することができる、マルチプロセッサシステム、例えば通信システムのマルチプロセッサシステムにおける負荷の平衡調整方法に関する。

【0002】

【従来の技術】 マルチプロセッサシステムがプロセッサ制御される通信システムであるとき、この通信システムは、予測される交換トラヒックおよび/または例えば接続形成時間(スルーフィー・コネクション・ディレイ)に課せられる厳守すべき最小要求に基づいて設計される。交換システムの設計を規定する重要なパラメータは、期待されるトラヒック値によって前以て決められる。その際のトラヒック値は、占有持続時間の和から一段測定期間における占有状態に関連して計算される商から得られる。しかし従来のトラヒック期待値に従って設計されるプロ

セッサ制御される通信システムはまた、特定の日、個々の時間帯または予め決定することができないイベントの際に発生するピーク負荷が生じた場合に到来するすべての交換問い合わせ(照会)をほぼすべて、例えば負荷平衡調整方法を用いて処理することができなければならぬ。

【0003】

【発明が解決しようとする課題】 本発明の課題は、若しい負荷状況においてもできるだけ僅かなプロセッサ要求しか拒絶されないことを保証する、リアルタイムシステムに対する負荷平衡調整方法を提供することである。

【0004】

【課題を解決するための手段】 この課題は、本発明によれば、それぞれのプロセッサがその負荷状態を定量化された量の形において求め、それぞれのプロセッサに、別のプロセッサの負荷状態をある時間バターンで通報し、それぞれのプロセッサが、当該負荷状態が所定の量を上回ることに依存しておよびその他のプロセッサの負荷状態に依存して、前記プロセッサに生じるタスクの少なくとも一部を前記残りのプロセッサに送出し、該送出されたタスクを、前記残りのプロセッサの負荷状態に相応し

(3)

特開平7-152591

3

て該残りのプロセッサに分配することによって解決される。

【 0005 】

【 作用】本発明によれば、プロセッサの負荷状態の考慮に基づいて、個々のプロセッサの過負荷が空いている、過負荷されていないプロセッサに分配され、その際分配されたタスクは、ネットワーク化されたプロセッサ内になんらの付加的な負荷変動を惹き起さずしかもこれによりスルーコネクションディレイの要件の充足、保持はなんの影響を受けることがないという利点が得られる。

【 0006 】さらに、本発明によれば、“限界負荷以下”にあるプロセッサが分配されたタスクによって過負荷されることはないという利点の他に、プロセッサの負荷状態はその“高負荷しきい値”を中心(指向)ないしオエインテドにされるという別の利点が生じる。

【 0007 】本発明の実施例によれば、負荷特性値にはヒステリシスが加えられ、このために、過負荷状態に入る際の過振動が回避されるという利点が得られる。

【 0008 】本発明の別の実施例によれば、付加的なタスクは、その負荷状態が“high”で段階付けられるプロセッサによっても処理され、これにより、プロセッサ毎に割付け指定される外部のタスクの数が使用可能なその空き容量に向けられるという利点が生じる。

【 0009 】本発明の実施例によれば、過負荷状態にあるプロセッサから分散分配すべきタスクの部分(関与成分)は負荷特性数から求められる。

【 0010 】本発明の実施例において、自ら処理される負荷に対して、分配される負荷がそれに対し推定されて加算される。これにより、推定され、実際にプロセッサに加わる負荷が高負荷しきい値以下に低下しつつ交換技術的な過負荷段階付けが率にセットされかつ保全期間が経過したよきようやく、プロセッサの過負荷状況が終了したものと見なされるという利点が生じる。

【 0011 】本発明の有利な実施例は、その他の請求項に記載されている。

【 0012 】

【 実施例】次に本発明を図示の実施例につき図面を用いて詳細に説明する。

【 0013 】図1には、本発明を理解するために必要な範囲において、通信システムKの構成要素が略示されている。これら構成要素は、加入者端末装置T_{n1}、…、T_{nm}、これら加入者装置に接続されている加入者接続線路の接続のための接続回路ユニットLTG₁、…、LTG_n、群プロセッサGP₁、…、GP_n、コオーディネーションプロセッサCPまたはコオーディネーションプロセッサの機能を引き受ける群プロセッサGP、別の交換システムに対するインターフェース装置SE、サービスユニットSU、非同期転送モードにおいて動作するマルチプロセッサAMX並びに非同期転送モードATMによって所望の加入者端末装置T_{nm}に対する接続路を通

10

20

30

40

50

し接続するスイッチ回路網SNBである。接続回路ユニットLTG_nはそれぞれ、複数の加入者端末装置を統合する。接続回路ユニットLTG_nに生じる交換技術的なタスクは、その都度少なくとも1つの群プロセッサGP_iによって処理される。その際通信システムKにおいて使用される群プロセッサGP_nは実質的にすべて同形式で構成されている。群プロセッサGP_nは、マルチプレクサAMXを介して通信システムKの引き続き処理すべきユニットにアクセスする。このマルチプレクサAMXに、群プロセッサGP_nの他に、サービスユニットSU、スイッチ回路網SNB並びに例えばコオーディネートプロセッサCPのような別のプロセッサも接続されている。非同期転送モードを介してマルチプレクサAMXはとりわけ別の交換システムにアクセスすることもできる。

【 0014 】群プロセッサGP_nは主に、接続形成および接続解除を引き受ける。これらはさらに、この種の交換技術的なタスクを処理するために、これらに接続されている接続回路ユニットLTG_nの固有の特徴的な特性に関する特性データを有している。群プロセッサGP_n間のデータ交換は、AMXを介して行われる。

【 0015 】図2には、群プロセッサGP_iの、本発明にとって重要な、2、3のシステム構成要素が示されている。群プロセッサGP_iは内部で、それぞれ異なったタスクを引き受ける複数のバーチャルプロセッサユニットVCPUに分割されている。“中央プロセッサユニット”C_iPUの計算ユニットの、その都度のマイクロプロセッサユニットVCPUに対する割り当ては、固有の割り当て方法(スケジューリング)を介して行われる。本発明との関連において、バーチャルプロセッサユニットVCPU-C_i1Pは特別重要である。というのは、それは電話接続の交換を引き受けるからである。バーチャルプロセッサユニットVCPU-C_i1Pは内部が、次の複数のプロセスによって構成されている：例として以下に2、3のプロセス並びにそのタスクを列記する：

—E S I S：拡張されたシグナリングおよび相互動作サブシステム

このEWSXシグナリングおよび相互動作サブシステムE S I Sは、接続回路ユニットLTG_nに属している種々の加入者端末装置T_{n1}、…、T_{nm}間のインターフェースを形成しつつ外部シグナリングシステムと内部メッセージインターフェースとの間の整合を実現する。

【 0016 】—E C P S：拡張された呼び出し処理サブシステム

このサブシステムは、呼び出し経路設定、呼び出しセットアップ、呼び出しクリアダウン並びに固有のフィーチャーの処理を引き受ける。このプロセスの処理のため、A T S (Access Transaction Segment)、U T S (User Transaction Segment)、A S (Associator Segment) …のサービス業務のような固有のサービス業務を

(4)

特開平7-152591

5

要求することができる。

【0017】これらプロセスは、相互に所謂メッセージを交換する。加入者端末装置Tn mによってトリガされる、接続形成のためのプロシージャの呼び出しの接続毎に、こここのプロセス間で複数のメッセージが送出される。個別例では100より多くの通報プロシージャから成るこの通報トラヒックは、呼び出し毎に唯一の群プロセッサGPIにおいて実行される。呼び出しが分配されると、“ホーム群プロセッサ”GPIに存在するプロセスESI SとECP Sプロセス並びに別の交換技術的なプロセスとの間のメッセージがATMネットワークを介してまだ“負荷可能な”群プロセッサG Pxにおいて実行される。

【0018】図3には、群プロセッサGPIに加えられる呼び出しCをまだ“負荷可能な”群プロセッサG Pxに転送する分配模式が略示されている。

【0019】バーチャルプロセッサユニットオペレーション&メインテナанс(高優先度)VCP U-OAMHによってその都度そのタスクを処理するためのプロセッサ時間指定されて得る、群プロセッサG Pnにおいて設計企画されているロードマネージャG PL Mが、群プロセッサの瞬時の負荷状態(群プロセッサロードストレート)GPL S並びに固有の交換技術的な過負荷防護段を指示する値OPL(Overload priority Level)を求める。この固有の交換技術的な過負荷防護段OPLはその都度、群プロセッサの負荷状態-“overload”において求められる。群プロセッサG Pnの次の負荷状態-normal, high, overloadおよびextreme-は、群プロセッサの中央プロセッサユニットCPUの負荷に相応して、ロードマネージャG PL Mによって求められる。その際群プロセッサG Pxの個々の負荷状態GPL Sは例えば次のように特徴付けられている：

GPL S-normal: 群プロセッサの実際負荷は、それに対して規定されている通常負荷しきい値以下である。通常負荷しきい値は例えば0.71 Erlangにある。(Erlangはトラフィック値の単位であり、その際0.71 Erlangは71パーセントの群プロセッサの実際負荷に相応する)

GPL S-high: 群プロセッサは、高負荷状態、すなわちその通常負荷しきい値より上方にある。この状態において、群プロセッサに対する過負荷防護措置はまだ開始されない。

GPL S-overload: 群プロセッサは過負荷状態にある。群プロセッサが処理すべきタスクは、出来る

だけ短期間の間に、過負荷されていない別の群プロセッサG Pnに伝送される。過負荷しきい値は例えば、>0.8 Erlangにある。

【0020】群プロセッサG Pnにおいて設計企画されているロードマネージャG PL Mは、負荷状態GPL Sを求めるこの他に、群プロセッサG Pnの負荷状態-overload-の期間に交換技術的な過負荷防護段OPLの決定も引き受ける。交換技術的な過負荷段OPL nは、群プロセッサG Pnが過負荷状態にある期間には交換技術的なプロセッサおよびそれに属する周辺機器によって実施することができる過負荷防護の強度に対する尺度である。その際交換技術的な過負荷段OPLは有利には、0と6との間の段階に分割される：

-OPL 0 防護なしを意味し、
-OPL 1 新たに到来する呼び出しの約16%が拒絶され、
-OP 6 優先付けられた接続路を介して別の通信システムから到来する呼び出しだけがまだ受容される。

【0021】ここに列記した、群プロセッサG Pnの負荷状態GPL S並びに交換技術的な過負荷段OPL nは、通信システムの設計に相応して任意に、精細化または大まかにすることができる。

【0022】群プロセッサG Pnが負荷状態“overload”になれば、呼び出しはそれ以上分配されない。しかし群プロセッサが負荷状態“overload”にあれば、0と1との間で変動する値に基づいて新たに到来する呼び出しの、空いている群プロセッサG Pxにおいて引き続いて処理するために転送すべき部分が決定される。残りの群プロセッサの引き続く呼び出しに対する受容能力を表す引き続き求めることができる値によって、通信システムKのこれら群プロセッサG Pnの全体の負荷状態に基づいて、概して分配することができるのかどうかが確定される。呼び出しを分配すべきことが決定されたとき、群プロセッサの求められた値pに相応して、ECP Sプロシージャを処理するためにどの“目標群プロセッサ”G Pxが準備されるべきであるかが決定される。

【0023】値qおよびpはそれぞれの群プロセッサG Pnによって負荷状態GPL Sおよびその固有の負荷値aによって局所的に計算される。以下に、値pおよびqを計算するための擬似コードを示す。

【0024】

【表1】

(5)

特開平7-152591

```
PROCEDURE calc q;
! calculate q;
BEGIN
  IF q=0
  THEN q:=min (1,freeGPs/overloaded GPs-0.1);

  IF myOLC.GPLS<2
  THEN BEGIN ! GP not overloaded;
    IF q<freeGPs/(overloaded:GPs+0.01)      ! q too small;
    THEN q:=MIN(1,free GPs/(overloaded GPs+0.01))
    ELSE q:=MIN(q*p sum,1); !no distr. ness., slowly
increase q
  END
  ELSE BEGIN !GP is overloaded;
    IF p sum > 1      !too much distributed;
    THEN BEGIN
      q:=q*p sum;
      damping:=damping+1;
    END
    ELSE BEGIN !more distribution possible;
      IF (myOLC.OfferedLoad>0.95 OR myOLC.OPL>0) AND
```

[0025]

[表2]

(6)

特開平7-152591

```

7
damping>2
THEN BEGIN
  q:=MIN(q*1.2,1);
  damping:=0;
END
ELSE damping:=damping+1;
IF myOLC.offeredLoad<0.65 AND myOLC.OPL=0
THEN BEGIN
  q:=MAX(0,q/1.1);
  damping:=damping+1;
  END;
END;
END GP is overloaded;
END PROCEDURE calc q;
END CLASS smartGPL TAB;

```

8

```

PROCEDURE calc p;
! calculate p(i);
BEGIN
  INTEGER i;
  REAL alpha;  !Initializationprob. for p;
  alpha:=1/{4*(freeGPs+0.1)};
  p_sum:=0;
  FOR i:=1 STEP 1 UNTIL NumberOfGPs DO BEGIN
    IF the GPL TAB(i)=0
    THEN BEGIN
      IF p(i)>0 THEN p(i):=alpha ELSE p(i):=p(i)*1.05;
    END;
    IF the GPL TAB(i)=1
    THEN p(i):=max(0,p(i)*0.9);
    IF the GPL TAB(i)>1
    THEN p(i):=0;
    p sum:=p sum+p(i);
  END;
  IF p sum>0 THEN BEGIN  (normalize p());
    FOR j:=1 STEP 1 UNTIL NumberOfGPs DO
      p(j):=p(j)/p sum;
  END
END PROCEDURE calc p;

```

【 0026 】 図4には、ある群プロセッサGP_jにおいて例えば局所的に発生する過負荷を処理するために呼び出されかつ処理される個々のプログラムプロシージャが示されているシーケンスがフローチャートの形で示されている。

【 0027 】 図3の分配模式図で示されていた値p_i(n)、q_j並びに負荷状態GPLS_jの計算は、上述したように、群プロセッサGP_jのプログラムモジュールロードマネージャGPLMにおいて行われる。GPLMは、バーチャルプロセッサユニットVCPU-OAM 50

Hによって処理されるプロセスである。

【 0028 】 次に、シーケンスフローチャートにおいて示されたステップについて説明する。

【 0029 】 一負荷値aの計算: 負荷値aは、それぞれの群プロセッサの中央プロセッサユニットの負荷から求められる。

【 0030 】 一群プロセッサGPsのロードステイツGPLSの決定: ロードステイツGPLSの決定のためには、“推定器”の形式において持続的な過負荷状況を検出する統合スタートインジケータS、負荷平衡調整フラ

(7)

特開平7-152591

9

GLBF並びに例えば値0..1が割り当てられることが
できるヒステリシス幅xが使用される。

【0031】定量化された量GPLSの決定は、次の擬似コードに従って実行される：

a) これまでのGPLS "normal"

if a > 通常負荷しきい値+x

then GPLS := "high"; *

c) これまでのGPLS "overload"

if LBF=true

then a' := a / (1 - q(j) d);

else a' := a;

if a' > 高負荷しきい値またはOPL > 0

then begin

GPLS := "overlast"

real overlast := true;

if (OPL = 6 since 3s)

then GPLS := "extrem"

end

else begin(ここからは真の過負荷はもはや存在しない)

if no rejection since 6s

then begin

if a' < 高負荷しきい値

then begin

GPLS := "high"

if a' < 通常負荷しきい値

then GPLS := "normal"

end;

end

else begin

GPLS := "overload"

real overload := false;

end;

end;

d) これまでのGPLS = "extrem"

固有の接続ユニットLTGsによって新たな負荷トライックはもはや認められない。

【0032】プログラムプロシージャ"GPLSの決定"は実質的に、最後の秒のプロセッサの負荷値aによって決定される。既に分配される場合、群プロセッサに実際に加わる負荷を表す量a'を介して計算される。負荷値a'は、次の監視期間の中央プロセッサユニットCPUに対して期待される負荷に対する、見積もりを表している。

【0033】すべてのGPnの負荷状態に関するのすべてのGPnに対するロードマネージャの通報

このために情報分配の2つの可能性が考慮される：

a) すべてのGPLMは、ロードコントロールマスター(LOCM)と称する中央集権化されたプロセスにおけるその局所的な付加情報(GPLS, OPL)を特定のプロセッサGPxに送出する。LOCMはこれら付加情

40

報を収集しあつできるだけ迅速に収集された情報をシステムにおけるすべてのGPLMに送出する。

【0034】b) GPLM間の'それぞれからそれぞれに通報する'交信を用いて、それぞれのGPLMはその局所的な付加情報(GPLS, OPL)をシステムにおけるすべての別のGPnに分配する。

【0035】すべてのGPLSの一時記憶の短い時間間隔の後、これらはすべてのGPnに分配されかつそこでその都度表に書き込まれる。短い時間間隔後、それぞれの群プロセッサGPxの更新されたGPLSは群プロセッサ表への書き込みの準備状態にある。

【0036】一付加的に処理すべきタスクp(i)の決定：その際値p(i)は、空いている群プロセッサGP(i)において処理されるべき、「ホーム群プロセッサ」から分配分配される呼び出しCの部分である。

【0037】その際値p(i)は、次の手法(ストラテジー)に従って決定される：

*if S > Thresh and a > 高負荷しきい値+x

then GPLS := "overload";

b) これまでのGPLS "high"

if a < 通常負荷しきい値

then GPLS := "normal";

if S > Thresh and a > 高負荷しきい値+x

then GPLS := "overload";

c) これまでのGPLS "overload"

if LBF=true

then a' := a / (1 - q(j) d);

else a' := a;

if a' > 高負荷しきい値またはOPL > 0

then begin

GPLS := "overlast"

real overlast := true;

if (OPL = 6 since 3s)

then GPLS := "extrem"

end

else begin(ここからは真の過負荷はもはや存在しない)

if no rejection since 6s

then begin

if a' < 高負荷しきい値

then begin

GPLS := "high"

if a' < 通常負荷しきい値

then GPLS := "normal"

end;

end

else begin

GPLS := "overload"

real overload := false;

end;

end;

d) これまでのGPLS = "extrem"

固有の接続ユニットLTGsによって新たな負荷トライックはもはや認められない。

【0032】GPLM間の'それぞれからそれぞれに通報する'交信を用いて、それぞれのGPLMはその局所的な付加情報(GPLS, OPL)をシステムにおけるすべての別のGPnに分配する。

【0035】すべてのGPLSの一時記憶の短い時間間隔の後、これらはすべてのGPnに分配されかつそこでその都度表に書き込まれる。短い時間間隔後、それぞれの群プロセッサGPxの更新されたGPLSは群プロセッサ表への書き込みの準備状態にある。

【0036】一付加的に処理すべきタスクp(i)の決定：その際値p(i)は、空いている群プロセッサGP(i)において処理されるべき、「ホーム群プロセッサ」から分配分配される呼び出しCの部分である。

【0037】その際値p(i)は、次の手法(ストラテジー)に従って決定される：

11
 a) $GPLS(i) = "normal"$
 $p(i)$ は高めることができる。というのは、 $GP(i)$ においてまだ処理容量を利用利用することができるからである。

【0038】b) $GPLS(i) = "high"$
 $p(i)$ は低減される。というのは、 $GP(i)$ が著しく多くの負荷を有しているからである。

【0039】c) $GPLS(i) = "overload"$
 $p(i)$ は常にセットされる。というのは、 $GP(i)$ が過負荷の作動状態にあるからである。

【0040】値 $p(i)$ の零調整設定の際に、空いている群プロセッサ $GP(i)$ 当たりの平均的な分配部分が 出力される。 $GP(i)$ が負荷状態 "overload" にある限り、 $p(i) = 0$ である。負荷状態が "overload" から "normal" または "high" に変化すると、値 $p(i) := 1 / (C * (\text{空いている } GP(x) \text{ の数}))$ が確定される。C は、立ち上がり振動フェーズにおける負荷分配システムの過振動の低減に関する交換特有の考察に 関わるパラメータである。

【0041】一分散分配すべきタスク $g(j)$ の決定
 群負荷プロセッサが負荷状態過負荷にあるとき、分散分配すべき呼び出しの部分を指示する値 $g(i)$ の前に、 $p(i)$ に基づいて、そもそも分配することができるかどうかが検出される。まだ正規化されていない $p(n)$ の和が 1 より大きな値にあるとき、値 $q(j)$ は、その都度の $GP(j)$ の要求に応じて決定される。負荷値 a に基づいて、 $q(j)$ を高めるべきかまたは低減すべきかが検出される。 $GP(j)$ が負荷状態 "overload" にない場合、常時、 $q(i)$ を高めることが試みられる。

【0042】a) 交換技術的な過負荷段 $OPL > 0$ または負荷値 $a > 0.8 + x$ (ただし $x = 0, 1$) である場合、 $q(i)$ は高められる。

【0043】b) $GPLS(j) = "overload"$ および $OPL = 0$ および負荷値 $a < 0.8 - x$ である場合、 $q(j)$ は低減される。

【0044】c) $GPLS(i) = "overload"$ である場合、 $q(i)$ を高めよ。

【0045】場合 c) における値 $q(j)$ の決定は、負荷分配への始まりに関連する。

【0046】その場合、突然の過負荷において最大の分配を得るために、普通は $q(i) = 1$ とすべきである。通信システム K がその出力限界において動作するとき、 $q(i)$ は、付加的な群プロセッサ GP_x が過負荷状態になることによって負荷分配を負かすことがないようにするために、 $q(i)$ は小さくなければならない。この理由から、 $q(i)$ は、個々の $p(n)$ の和が 1 より小さい場合、 $q(j)$ は低減される。

【0047】過負荷過程での生じ得る振動を回避するために、値 $q(j)$ の低減の前に、短い時間間隔だけ待つ

(8) 特開平7-152591

12

ようになると、有利である。群プロセッサ $GP(j)$ が呼び出しを分配しようとしていることが検出された後、負荷平衡調整フラグ LBF "true" がセットされ、そうでないときは、 LBF "false" がセットされる。ガード期間と称される保護時間間隔においては、群プロセッサ $GP(j)$ が負荷状態 "overload" にあるかもしれないが、負荷平衡調整フラグ $LBF := false$ がセットされる。

【0048】値 $q(i)$ の新たな設定の際に、次のアルゴリズムが使用される: $q(j) = 0$ である場合、 $q(-j) := \min(1, (\text{空いている } GP_x \text{ の数} / \text{負荷状態 "overload" にある } GP_s \text{ の数}))$ 。 $GP(j)$ の $GPLS$ が "overload" にならかつ $q(j) < (\text{空いている } GP_s \text{ の数} / "overload" \text{ にある } GP_s \text{ の数})$ であれば、 $q(j) := \min(1, (\text{空いている } GP_s \text{ の数} / "overload" \text{ にある } GP_s \text{ の数})$ がセットされる。 $GP(j)$ が "overload" にないときであって、 $GP(j)$ が突然過負荷状態に陥った場合、負荷分配への迅速な開始を保証するために通常は $q(j) = 1$ である。しかし外部の過負荷を引き受ける準備がある GP_s より多くの GP_s が "overload" にある場合、 $q(i)$ は、過負荷にならない GP_s を過負荷から防護しかつひいては負荷分配システムを損なうことがないようにするために、適合的に比較的小さく選択される。

【0049】呼び出しの分配は次に示すように行われる: 呼び出しが生じると、処理すべき呼び出しに対する直接的なインターフェースでありかつ呼び出しの本来の分配を行う サービスマネージャ SM が LBF に基づいて、分配すべきか否かを検出する。分配すべきである場合、生じた呼び出しを分配することができるかどうかが検査される。分配が基本的に可能でありかつ所望される場合、この固有の呼び出しを分配すべきであるかどうかが判定される。このために、所定の時間間隔においてこれまで分配された呼び出しと到来するすべての呼び出しの数とのから成る商が形成されかつ $q(i)$ と比較される。この商が $q(i)$ より大きければ、この呼び出しは分配される。例えばこれまで到来した呼び出しの数が 20 であり、これまで分配された呼び出しの数が 4 でありかつ $q(i) = 0.25$ であれば、商 $4 / 20$ から、この商が $q(j)$ より小さいことがわかる。この場合、呼び出しは分配される。呼び出しを分配すべきであるとき、これに類似して、 $p(i)$ の値に基づいて、どの群プロセッサ GP_x が呼び出しの処理を引き受けるかが判定される。

【0050】

【発明の効果】本発明によれば、プロセッサの負荷状態の考慮に基づいて、個々のプロセッサの過負荷が、空きのある過負荷されていないプロセッサに分配されて、分配されたタスクが、ネットワーク化されたプロセッサ内になんらの付加的な負荷変動を惹き起こすことなくしか

50

もこれによりスルーコネクションディレイの充足、維持はなんの影響を受けることがないという利点が得られる。

【図面の簡単な説明】

【図1】通信システムの部分を示す概略図である。

【図2】図1の通信システムに属する群プロセッサのプロセス処理ユニットを示す概略図である。

13

(9)

符開平7 -1 5 2 5 9 1

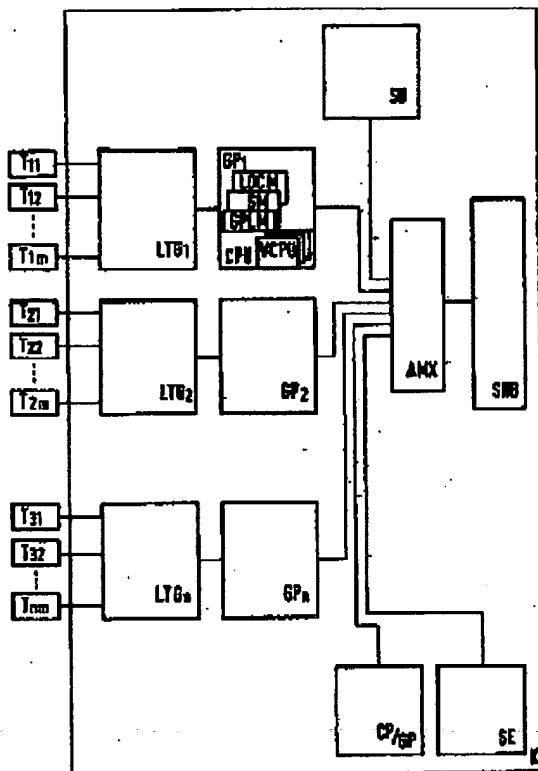
14

【図3】本発明の方法を説明する模式図である。
【図4】シーケンスを説明するフローチャート図である。

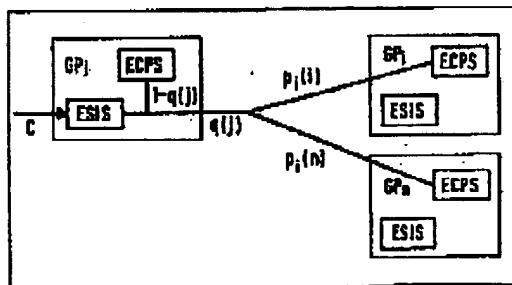
【符号の説明】

K 通信システム、 GP プロセッサ、 GPI.S
負荷状態

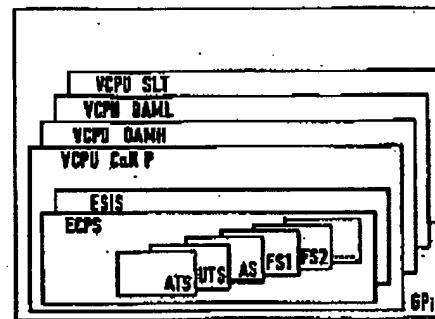
[図1]



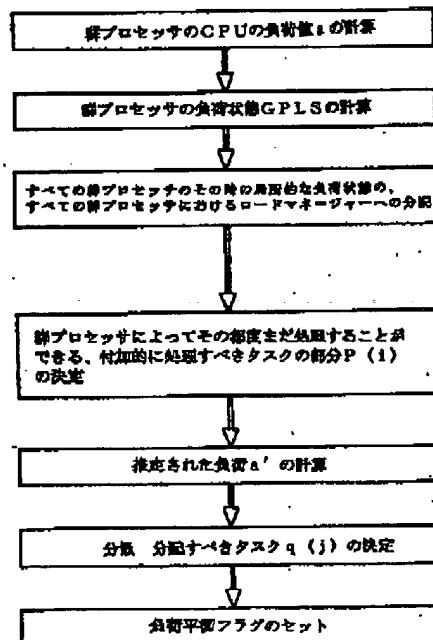
[圖3]



[図2]



[图4]



(1 0)

特開平7-152591

フロントページの続き

(72)発明者 ヴァルター ヘルト
ドイツ連邦共和国 グレツリート イーザ
ーダム 129